

Monitoring and Analysis of Data from Complex Systems

Thomas Dollman and Kenneth Webster
National Aeronautics and Space Administration
Marshall Space Flight Center

As flight systems become more complex and longer-lived, it becomes increasingly difficult to monitor and analyze their data. One method being developed to address this problem is the use of ground-based information systems in ways that enable the engineer to relate the data to the system design more readily. Another method is to encode knowledge about the system's operation into the ground computer system itself, to serve as a backup to the engineer's analysis and conclusions. These methods, now being tested at the Marshall Space Flight Center, promise to help maintain the high productivity levels of the telemetry analyst over long periods.

INTRODUCTION

Complex spacecraft, such as the Hubble Space Telescope (HST), are monitored closely and continuously during flight. The initial mission phase of the HST, the Space Telescope Orbital Verification (STOV), was supported by engineers, scientists and managers at MSFC's Huntsville Operations Support Center (HOSC) and at Goddard Space Flight Center's Space Telescope Operations Control Center. Data originating from the HST is being constantly monitored and analyzed by both computers and humans to determine how well this spacecraft is operating. Fault Diagnosis, Isolation and Recovery (FDIR) is the driver for much of the telemetry monitoring and analysis activities. Ground-based facilities have been receiving and processing over 5,000 telemetry measurements per data-transmission cycle from the HST continuously since its launch in April, 1990.

This paper provides an overview of the computational infrastructure in the HOSC which enables the telemetry monitoring and analysis functions for both spacecraft and vehicles to determine their status and to support corrective actions, when required. Also discussed are some new methodologies and tools that have been employed for on-line monitoring and analyses of telemetry measurements, as well as systems that empower the analysts with off-line tools for information retrieval. To provide perspective on the mainline HOSC computational infrastructure, both historical information and an extensive description of work-in-progress are discussed. These currently used and evolving information systems enable the engineering analysts to effectively perform those functions required for ground-based support of flight systems. Some of the software elements described herein may be applicable to other domains wherein complex systems are being monitored and/or analyzed. All elements are described at various levels of functional detail to enable the reader to decide where the likelihood for such applications might exist.

HISTORY

In the late 1960's the HOSC consisted of a data acquisition system (Honeywell DF224) and a data presentation system (Burroughs B5500 LP) to support initial Saturn IB launches. The first upgrade, for late Apollo missions, was to the data presentation system. This system was upgraded to a Univac 1108 driving 8 digital display devices, and served multiple users through a video-switching system. The system was somewhat interactive in that display images could be dynamically assigned to any of 8 display channels. Software was mission-dependent and since launches were infrequent, this allowed the software to be altered for subsequent missions. The next upgrade was to prepare for Shuttle missions by using Perkin-Elmer systems driving both 10-channel and 8-channel display systems simultaneously. The major new feature was the ability to define displays dynamically and the assignment and scaling of data to lights, meters, and strip-chart recorders. This enhancement was required for increased launch activities requiring rapid reconfiguration. This system also supported simultaneous real-time presentation and near real-time data recall from data bases. However, definitive data reduction was still relegated to off-line, post-mission data processing activities.

CURRENT INFRASTRUCTURE

In the early 1980's, design began on the next generation of HOSC data display systems which evolved into the currently implemented Peripheral Processing System (PPS). The purpose of the PPS is to serve as the backbone of the Payload Operations Control Center (POCC) in the HOSC, and it was targeted to specifically support STOV. As an offshoot, the PPS was incorporated into the HOSC's Space Shuttle launch support system as well. The PPS microVAX/Perkin-Elmer based system is networked together on an Ethernet. The Perkin-Elmer systems (which serve as Central Processors - CPs) provide centralized functions, such as data acquisition and decommutation, and the microVAXes (Peripheral Processors - PPs) provide distributed functions, such as data display. In addition to its data display capabilities, the PPS also includes the ability to control payloads from the ground. The primary purpose of the PPS is to provide short-term support for shuttle missions carrying a wide variety of payloads. Thus, for quick turn-around between missions, the PPS was designed to be reconfigurable without changing existing software. Since its inception, the PPS has proved to be very flexible in its support of many types of missions/activities.

The CPs perform centralized functions common to the entire system. The primary function of the CPs are to receive telemetry data in downlink format and provide a pathway for commanding remote payloads. The telemetry data is made available to the PPs in Real-Time (RT) and Near-Real-Time (NRT) modes. NRT data is data from the near past that is saved in data bases on the CPs. The amount of NRT data that can be saved is dependent on both mission and hardware complement. The CPs also monitor the telemetry data for exception conditions that are mission-critical and generate messages that are sent to the PPs to advise the users when exceptions occur. In addition, the CPs maintain and distribute to PPs data base parameters.

The PPs provide distributed functions that are user-specific. On a PP a user can build and modify, in real-time if necessary, free-form displays for viewing his RT or NRT telemetry data. The display capabilities include: tabular fields, X-Y plots, limit sensing, and some graphical representation. When the display is executed, the PP requests data base information from the CPs about each of the parameters to be displayed. Based on that information, it then requests, receives, processes, and displays those parameters from the CPs. In addition, the user may build Special Computations for any special processing that required to be performed on telemetry data. A Special Computation is a FORTRAN program which provides algorithms for processing the data. A package of routines is provided for retrieving data and making any derived parameters available for display. The user may build a Special Computation directly with a text editor or he may do so indirectly with a set of forms that is provided. The user provides a list of input parameters, the FORTRAN algorithms for processing those parameters, and a list of derived output parameters. The forms program then generates the code necessary for retrieving/writing the parameters.

In summary, the Peripheral Processor System provides an effective way to simultaneously provide telemetry information to a number of users in the HOSC. It enables users to build, store, retrieve and modify/display pages to monitor RT and NRT data in columnar and graphical form. However, in order to evaluate new techniques of viewing data (and documentation), a number of different systems were tested both on-line and off-line in the HOSC during STOV.

NEW ON-LINE SYSTEMS ARE TESTED

Considerable design and engineering knowledge about a spacecraft is documented at various points during its lifecycle. The future availability of this information is important to consider in light of the fact that spacecraft like the HST may require a decade or longer from design to launch, and are utilized on orbit for perhaps two more decades. A life-cycle three decades long or longer, as in the case of the Space Station Freedom, precludes the direct involvement of systems and subsystems designers during much of the operations phases. Therefore, systems which enable or aid the FDIR processes by providing fast, accurate ways to retrieve the specific knowledge pertinent to a specific problem need to be available to help engineers maintain high levels of spacecraft functionality throughout its operational lifetime.

During the STOV, a number of new concepts for telemetry monitoring and analysis were tested in the HOSC. Three of these systems, the HST Operational Readiness Expert Safemode Investigative System, the Vehicle Health and Safety Expert System, and the Thermal Control Subsystem Expert System were implemented in a Knowledge Based Systems (KBS) Testbed, and were operated as an adjunct to the standard PPS functionalities by using a High-Speed Peripheral Processor. Each of the three KBSs provides the user with a unique method for relating telemetry values to the status of the HST.

High-Speed Peripheral Processor

Data for the KBS Testbed was provided by a High-Speed Peripheral Processor (HSPP), which was interfaced to the Central Processors as a standard Peripheral Processor utilizing configuration-controlled PP software and four Special Computations. The HSPP was implemented as a VAXServer 3300 using VMS and special-purpose interface software elements. The HSPP collected 1400+ telemetry data-items per data-cycle from the CPs and distributed this data over the isolated KBS Testbed network, utilizing TCP/IP, to the three KBSs described below.

HST Operational Readiness Expert Safemode Investigative System

A reusable knowledgebase shell, the Device Reasoning Shell (DRS), elaborated upon below, was built to access a variety of device models and rule bases to allow a user to solve a variety of problems. To test and demonstrate the application of DRS to the HST, a DRS application was developed which provides assistance in the investigation of HST system anomalies, which are called safemode events [1]. The application, the HST Operational Readiness Expert Safemode Investigation System (HSTORESIS), contains rule bases and a device model that encode knowledge about HST safemode detection, fault isolation, and recovery. A safemode event occurs when any one of a number of sequences of on-board measurements are determined to be off-nominal by the on-board Data Management Subsystem (DMS). The DMS then commands the spacecraft to enter a "safe" or protected mode. These on-board measurements are also available to ground support personnel, so that they can analyze and fix any problem from the ground, prior to uplinking commands to return the HST to normal operating mode. Therefore, it is necessary to thoroughly understand the causes of the sequence(s) of off-nominal measurements which resulted in the triggering of the safemode action. Procedural knowledge was obtained for HSTORESIS from HST safemode system and flight software documentation, and from HST design engineers. The HSTORESIS implementation provides the user with assistance in the investigation of safemode events. HSTORESIS runs as an application using the Device Reasoning Shell (DRS) on a Sun 4/260 workstation.

Device Reasoning Shell

The Device Reasoning Shell (DRS) is a reusable knowledgebase shell which can access a variety of device models and rule bases to allow a user to solve a variety of problems [2]. DRS was built using Common Lisp and Intellicorp's Knowledge Engineering Environment on a Symbolics 3670 and Texas Instruments' microExplorer before installation on the Sun 4. The DRS is designed to assist in solving problems that require the ability to reason about the model of a device. Knowledge about a device is captured within DRS rule bases and device models. Two programming paradigms, object-oriented programming and rule-based programming, are used to reason about device models. Interaction between the user and DRS is accomplished through a point-and-click style of interface, using buttons and windows. The DRS has three layers: Telemetry Interface, Model Manager, and User Interface. In addition, a data manager records all telemetry items for later replay and analysis, and distributes data items throughout the system.

The telemetry interface. The main function of the telemetry interface is to convert raw telemetry into a level of abstraction which is closer to the mental representation that human experts use. This telemetry passes through the telemetry interface and is categorized into one of several different telemetry types to be utilized by the other parts of the DRS. For the HST, 5,500 telemetry items are the main source of information about the behavior of the spacecraft. For the HSTORESIS application, each of the 234 different safemode-specific telemetry data items is mapped into its category (polynomial fit, counter, table lookup, bi-level, or multi-level), and then associated with information about that type.

The model manager. The model manager manages Device Models. A Device Model in DRS includes a mapping between the model and a set of monitors, pointers to rule-bases that are capable of reasoning about the device, behaviors that represent the conceptual or physical functioning of the device or component, and features that hold state information that is not included in the satellite telemetry stream. For HSTORESIS, the device model provides a model of the behavior of a device as a function of telemetry data. Devices include physical items like Reaction Wheel Assembly or Solar Array Assembly on the HST.

The user interface. The user interacts with DRS through a desktop. The user may point to and click on buttons which activate functions within the DRS application. These functions include startup & shutdown, selection of a message or data-item displayed in a window for elaboration & explanation, moving & resizing windows, and selecting window types for display of telemetry data in different formats. For the

HSTORESIS application, the user can create a number of "monitor pages" or screens-full of data displaying telemetry items per screen-full in formats which include actual value, strip-chart recorder, and/or X-Y plotter displays. Additionally, safemode events detected by HSTORESIS enable access to pop-up windows which describe causes, effects, and recovery procedures relating to the event(s).

Vehicle Health and Safety Expert System

The Vehicle Health and Safety (VHS) Expert System was developed at Lockheed Missiles and Space Company (LMSC), using the LMSC-proprietary L*STAR real-time expert system tool-set. The VHS system was implemented and tested in the KBS Testbed during STOV using a beta-test version of a commercially available real-time expert system package, Talarian Corporation's R*Time. This software was implemented on a VMS-based VAXStation 3100 and provided a sophisticated monitoring strategy of three different HST flight modes representing start-up (deployment), normal operations, and anomalous behavior (safemode) [3]. The workstation display, utilizing VI Corporation's DV Draw, provided telemetry monitoring graphics as follows.

Four graphs are displayed at the top of the workstation screen which summarize the pointing control and electrical power health of the spacecraft, and allow an observer to spot adverse trends as they develop. Spacecraft mode changes are color-coded, as are checklists of functions to be performed. If some activity is delaying a checklist function, the user may expand the list to observe telemetry values pertinent to that checklist item. In normal mode, display emphasis is on simultaneous status summaries of all pertinent spacecraft subsystems which are required to maintain health and perform the intended mission, the successful acquisition of science data. Safemode is autonomously entered by the spacecraft when any of several dozen reasonableness tests fail. That is, the on-board computer monitoring those sensor readings senses a problem. At that point, the important sensor readings are captured in a recorder, for later analysis by ground station. The ground-based VHS system also captures this data and performs analyses on it, provides status information for each data item, and recommends recovery procedures. The VHS therefore provides the user with a highly flexible monitoring station containing the procedural knowledge related to entering and recovering from anomalous behavior.

A special interface between the VHS Expert System and HSTORESIS was also developed and tested during STOV. The VHS system continuously monitored data pertaining to safemode events, and upon detecting such an event, VHS notified HSTORESIS that an event had occurred. At that point, the HSTORESIS user could access the pop-up windows and then perform an analysis on the pertinent subset of telemetry data. The analysis, via interpretation of strip-charts or X-Y plots, can be performed by replaying, pausing, and "stepping" through any of the previously stored telemetry signals for any portion of the 5 orbits (450 minutes) leading up to that event.

Thermal Control Subsystem Expert System

The Thermal Control Subsystem (TCS) Expert System was also developed at LMSC and utilizes identical development and delivery software and hardware as the VHS system [3]. This system was also implemented and tested in the KBS Testbed during STOV, and directly supported HOSC-based thermal engineers throughout STOV. Sensors on the HST enable ground-based FDIR activities on the TCS via some 1400 different data values. These sensors measure the temperatures of the HST internal and external structure as well as mechanical, electronic, and optical equipment (for overheating or overcooling alerts or warnings).

The TCS system contains several hundred solid images of spacecraft subsystems, components, and structures on line which are arranged in a hierarchical fashion. The user can "navigate" through the hierarchy by replacing a current display with a subwindow that is higher or lower in the hierarchy, thereby going from system to subsystem to component in a point-and-click fashion. Each image shows pointers to the location of its sensor(s) in a spatial fashion with real-time readouts of those sensors from the telemetry data. This spatial representation is especially helpful when considering the spacecraft's position with respect to the heating and cooling effects of the space environment. Should a telemetry value achieve an off-nominal value (too hot or too cold), the indicator value changes color depending on the severity. Also changing color are pointers that guide the engineer through the image hierarchy. Displayed on all image levels are warning/alert windows which report on the number, severity, time, and name of the off-nominal sensor(s), and which enable the user to directly access the image where each off-nominal sensor is located.

Additionally, the user can request, in an overlay window, a real-time plot of any sensor readings with respect to time. The TCS Expert System, implemented with R*Time and DV Draw, provides a robust, comprehensive telemetry monitoring system for the extensive Thermal Control Subsystem of the HST.

OFF-LINE AIDS TO ANALYSIS

Information systems which do not directly access telemetry data can also be extremely useful in evaluating that data. Decade-long design/development phases produce a wealth of design and engineering documentation which must be accessed quickly during operations phases. Two approaches to this access are described below. One system, the Document Retrieval Assistant, was utilized during STOV to enable fast access to stored documentation on the design, engineering, and operational aspects of a key part of the HST, the Orbital Telescope Assembly. The second tool, the Design Alternatives Rationale Tool, was developed to capture design knowledge about the Space Station Freedom during the engineering trade-study process, and is beginning to be utilized for fault analyses as well.

Document Retrieval Assistant

Background. The process of designing, developing, and testing any complex system or subsystem generates a large number of documents to describe the system and processes involved. In the case of the HST, the document-set size and scope is formidable, numbering in the tens of thousands of documents [4, 5]. Documents for programs are typically organized hierarchically, as in a "document tree." Systems are represented as collections of subsystems; subsystems are represented as collections of components. At each level, these elements interact, intercommunicate, and/or interconnect. In order to focus on particular elements or characteristics of a system, to learn or to refresh one's memory, quick access to all aspects of that element in that documentation-set is often required. These documents are often stored electronically, as bit-mapped images and/or text. The challenge is to find all pertinent facts about the subject, including interfaces, in an efficient manner.

Objective. To meet this challenge, a system has been developed for MSFC by Hughes Danbury Optical Systems called the Document Retrieval Assistant (DRA). This system serves as a complement to the Hubble Space Telescope Management Information System (TMIS) which contains, on optical discs, bit-mapped images of all drawings, specification documents, and top-level reports and memos relative to the HST. The DRA contains audio interviews, videotapes, and intelligent pointers to TMIS contents as relates to the Optical Telescope Assembly (OTA), a key part of the HST. The DRA was needed to provide the information to support OTA performance analysis and subsystem troubleshooting, repairing and reverifying certain defective sensors returned from orbit, and developing next-generation designs and modifications for certain sensors. The DRA was also developed to capture and file away OTA expert knowledge for retrieval of critical information when required, and for training and familiarization of personnel.

Using the DRA. The initial DRA system is fielded on an 80386-based workstation with windows-based user interface and strong emphasis on graphics. The DRA has four main components: hyper-text organization of documents, intelligent search, databases, and system administration and authoring tools. The DRA permits rapid search and retrieval of a very large collection of program documentation (text, photos, drawings, and databases) without resorting to keywords. The two major methods of retrieval are by document browsing (casual or in-depth) and document search (exact text matching, conceptual matching, and fuzzy matching). In short, the DRA is a tool which provides the user with an efficient tool to find all references to a particular component or process, and shows the user drawings of devices or abstracts of pertinent documents, as well as pointers to audio/video discussions and specific documents within the TMIS or other archives for retrieval of more in-depth information.

Design Alternatives Rationale Tool

One important aspect of completely understanding complex systems is the need to identify and characterize all possible failure modes. For instance, Failure Modes and Effects Analysis (FMEA) is performed on each of the Space Station's critical subsystems. It is important to ensure that all failure modes have been identified and their full effects are understood, including effects in other subsystems. Much later, in an operational environment, the analyst-team must correctly interpret a fault and understand its consequences before corrective action is taken. One tool, the Design Alternatives Rationale Tool (DART) can be used now by reliability engineers engaged in the FMEA process to build lists of the potential faults that may occur and relate those faults to different failure modes. The methods used in DART for the development, storage, and access of FMEA information should make it useful for the analyst to use during operations.

DART was originally designed for the Space Station Freedom Program as a tool to aid in the engineering analysis process and to easily capture rationale about the alternatives evaluated during its use [6]. It is useful for applications such as trade studies, aiding and documenting decisions made in meetings, and group data gathering. DART employs repertory grids as one form of knowledge representation. This representation appears to the DART user as matrices and allows the critically important capture of rationale behind the axis-entries. DART works well on analysis problems or those problems whose solutions can be enumerated (e.g., classification, interpretation, and diagnosis). Analyses of failures of complex systems fall into this category, so the DART tool is beginning to be to aid the FMEA process. DART was originally developed on Sun and VAX/VMS computers, but MacIntosh, IBM PC, and IBM mainframe versions are also available.

INFRASTRUCTURE EVOLUTION: THE MARSHALL INTEGRATED SUPPORT SYSTEM

An effort is currently underway to replace the current PPS in the HOSC with a state-of-the-art system. However, due to resource constraints and the need to retrain users, it is not feasible to replace the entire PPS in a single step. Thus, it is necessary to follow an evolutionary path where new technology is phased into the old system over a period of time. This allows users to become familiar with new technology while still using the older technology and spreads the cost of development and upgrade over time. Systems such as those described above provide insight and ideas for the evolution of this computational infrastructure.

Overview

As a proposed first step along this evolutionary path, the Marshall Integrated Support System (MISS) is being developed to support the Payload Operations Control Center (POCC), a HOSC infrastructure for support of shuttle payload elements. The main requirement of the MISS is to produce, by mid-1991, a state-of-the-art system capable of performing the current PP functions on high-resolution graphics workstations in place of the current DEC microVAXes. Except for data base services, the functionality of the CPs will not be modified until a later date. The system is being designed to support the Space Station Freedom and Advanced X-ray Astrophysics Facility, as well as the current POCC. In addition, it will provide a migration path for utilizing workstation technology throughout the HOSC.

Like the current PPS, the MISS will support many types of missions without the need for software modification. Although the functionality of the MISS is the same as that of the PPS, the MISS is quite different in that it takes advantage of graphics workstation technology. The system is implemented in C under the UNIX operating system and uses MOTIF for its look and feel. The major advantages of the MISS over the current PPS are its primary design goals, modularity and portability.

The functional design of the PPS is very interdependent, whereas, the MISS design is modular, so that each of its components is as independent as possible from all of the other components. In some cases, the components can be used entirely independent of the MISS. In addition, the modularity is further enhanced with packages of routines that are designed to isolate internal and external interfaces and commercial software packages. The modular design makes it easier to integrate enhancements and newer technology into the system. It also helps during development, because it is easier to test the different modules before integrating them with the rest of the system. In addition, modularity makes it possible to integrate commercial off-the-shelf (COTS) products into the system. For instance, if a user in a future mission has a requirement for a COTS product to analyze data, it will be simpler to interface that product to the system.

The current PPS is written in VAX-specific FORTRAN and is practically . The MISS is designed to achieve portability by using industry standards and by implementing the system on multiple platforms. The standards chosen for the system, UNIX, C, X-Windows, and MOTIF were chosen because the graphics workstations initially targeted to become MISS platforms will support them. This helps to enforce portability in that the platform dependency areas can be identified and isolated early in the project.

The functionality of the MISS breaks down into the following subsystems: Data Processing, Data Display, and Payload Commanding. The Data Processing subsystem allows the user to define a set of parameters to be retrieved from the CPs and the processing that is to be performed on those parameters. The Data Display subsystem allows the user to build graphical windows using a wide variety of textual and graphical methods to represent his data. The Payload Commanding subsystem provides a mechanism for the user to control his payload. As indicated above, each of these subsystems is modular.

Data Processing Subsystem

The Data Processing subsystem breaks down into three tasks: a Data Set Builder, a Data Processing Precompiler, and a Data Retriever. The Data Set Builder is the user friendly windowing interface for defining a set of parameters to be retrieved from the CPs, the processing to be performed on those parameters, and a set of derived parameters to be made available to other processes. The definition is used to create a text file of pseudo code which the Data Processing Precompiler translates into C code. The precompiler then executes the C compiler to compile and link the C code into an executable. This executable receives the raw telemetry parameters from the Data Retriever, converts and calibrates those parameters, executes the user specific processing, and makes the derived output parameters available through UNIX socket connections to the Data Display and the Payload Commanding subsystem. If there are insufficient resources in a single workstation to accommodate all of the subsystems, the socket connection allows the Data Processing subsystem to be moved across a network to another workstation, a further demonstration of the flexibility of the system. The precompiler also creates two additional files: a parameter request list which the Data Retriever uses to request parameters from the CPs and a data set list which describes the derived parameters that are available to the other subsystems.

The Data Processing subsystem combines the Special Computation, type conversion, and calibration functionality of the PPS into a single module. In the PPS data conversion and calibration are a part of the data display capability. In the MISS type conversion and calibration are assumed to be a subset of data processing. The Data Display subsystem deals only with data representation (not with data processing).

Data Display Subsystem

The Data Display subsystem breaks down into two tasks: the Display Builder and the Display Driver. The Display Builder is the user interface for defining the way, either graphical or textual, that data is to be represented on the screen. The Display Driver makes socket connections to the Data Processing subsystem to acquire data which it uses to update the user's display that he created with the Display Builder. This subsystem is the most flexible in that it is designed so that the Display Builder may be replaced with any appropriate COTS data representation tool. This required that the Display Driver be tool-specific and must be implemented to support the specific tool that is used. The Transportable Applications Environment Plus (TAE+) has been chosen as the default data representation tool. Other COTS products are being evaluated in order to demonstrate the flexibility of this subsystem.

Payload Commanding Subsystem

The Payload Commanding subsystem breaks down into two tasks: the Command Interpreter and the Command User Interface. The Command Interpreter implements the MISS System Test and Operations Language (MSTOL) which is the language through which users may control their payloads. The MSTOL includes features for modifying and uplinking commands, for accessing telemetry data, and for performing conditionals and loops. The Command Interpreter will accept MSTOL directives interactively or from a script file. The Command User Interface is the user-friendly windowing interface which allows the user to interact directly with the Command Interpreter, query the Command Database, and edit MSTOL script files. The MSTOL represents a significant improvement over the current PPS. The PPS provides mechanisms for modifying and uplinking commands, but does not provide a scripting language for defining logic to control a payload.

External Interfaces

A main menu interface is provided to allow the users access to the various subsystems within the MISS. The main menu, along with the interfaces to each of the subsystems, use the X Window System as well as the Open Systems Foundation MOTIF user environment to provide a consistent behavior throughout the MISS application. From the main menu, the user has access to a variety of applications including building and driving displays, commanding, building data sets, viewing the telemetry database, and accessing the UNIX operating system. Taking advantage of workstation capabilities, the user may access any number of these applications simultaneously and as often as workstation resources will allow.

Two other PPS components that are being replaced are the Command and Telemetry Databases. Each of these databases are custom, in-house databases which describe the telemetry parameters that are available and the commands that may be uplinked. These databases are being implemented with a commercial Relational Database Management System (RDBMS). RDBMSs allow new information to be added to the databases as it is required thus adding to the flexibility of the new databases.

SUMMARY

This paper has introduced some of the methods, systems, and prototypes that have been tested for monitoring and analyzing the data from several spacecraft and vehicles at the Marshall Space Flight Center. For the HOSC infrastructure, the MISS provides a migration path to the state-of-the-art workstation environment. Its modular design makes it possible to implement the system in stages on multiple platforms without the need for all components to be in place at once. In summary, the MISS provides a flexible, user-friendly environment for monitoring and controlling orbital payloads. In addition, new capabilities and technology may be incorporated into MISS with greater ease. The use of information systems technology in advanced prototype phases, as adjuncts to mainline activities, is useful to evaluate new computational techniques for monitoring and analysis of complex systems. Much of the software described in this document (specifically, HSTORESIS, DRS, DART, elements of the DRA, and software for the PPS and the HSPP) is available with supporting documentation from the authors, and may be applicable to other systems monitoring and analysis applications.

REFERENCES

- [1] Cox, P. A.; and Forbes, J. H.: An Application of Design Knowledge Captured from Multiple Sources. Proceedings of the Fifth Conference on Artificial Intelligence for Space Applications, NASA CP-3073, 1990, pp. 291-300.
- [2] Cox, P. A.; and Forbes, J. H.: Hubble Space Telescope Design/Engineering Knowledge-base Device Reasoning Shell (HSTDEK-DRS) User's Manual. LMSC/F373327. Lockheed Missiles and Space Company, 1990.
- [3] Sampedro, P. S.: Hubble Space Telescope Real-time Expert Systems. Unpublished. Lockheed Missiles and Space Company, 1990.
- [4] Clapis, P. J.; and Byers, W. S.: Intelligent Search and Retrieval of a Large Multimedia Knowledgebase for the Hubble Space Telescope. Proceedings of the SPIE Conference on Applications of Artificial Intelligence VIII, vol. 1293, 1990, pp. 590-593.
- [5] Clapis, P. J.: Optical Telescope Assembly Project Report: OTA Document Retrieval Assistant Prototype. PR-1423. Perkin-Elmer, 1989.
- [6] Boose, J.; Chew, S.; Shema, D.: DART User's Guide (Draft). Boeing Computer Services, 1990.